

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Farn	§	
	§	Group Art Unit: 2173
Serial No. 10/782,498	§	
	§	Examiner: Shrestha, Kiran K.
Filed: February 19, 2004	§	
	§	
For: Method and System for Editing	§	
Column Oriented Programming	§	
Language Statements		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on October 8, 2007.

A fee of \$510.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-28

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: none

Claims withdrawn from consideration but not canceled: none

Claims pending: 1-28

Claims allowed: none

Claims rejected: 1-28

Claims objected to: none

C. CLAIMS ON APPEAL

The claims on appeal are: 1-28

STATUS OF AMENDMENTS

No amendments were filed after the Final Office action of July 10, 2007.

SUMMARY OF CLAIMED SUBJECT MATTER

A.1. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method for editing a plurality of column oriented programming language statements presented to a user on a display screen. (Specification, page 2 lines 15-25) The method includes, identifying a template description for at least one of the plurality of column oriented programming language statements, (Figure 4, Figure 5, reference numeral 410; specification page 1 lines 6-7, page 16, lines 17-26) wherein the template description defines at least one statement type, (Figure 2, reference numeral 221, specification page 2 line 18, page 11 lines 4-21) and wherein the at least one of the plurality of column oriented programming language statements is selectable (Figure 2, reference numeral 203, specification page 11 lines 18-21) in a first pane of a plurality of panes on a graphical user interface (Figure 2, reference numerals 200, 210, specification, page 10, lines 13-27); and displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements (Figure 2, reference numerals 222, 224, 225, specification, page 11 lines 4-21).

B. CLAIM 8 – INDEPENDENT

The subject matter of claim 8 is directed to a system for editing a plurality of column oriented programming language statements, (Specification, page 3 lines 15-20) the system having a display and an input device for editing the plurality of column oriented programming language statements through a graphical user interface. The system comprises (Figure 1, specification page 5 line 12 through page 10 line 11) a processor coupled to the display and the input device and the processor adapted for identifying a template description for at least one of the plurality of column oriented programming language statements, (Figure 4, Figure 5, 410; specification page 16, lines 17-26) wherein the template description defines at least one statement type, (Figure 2, 221, specification page 2 line 18, page 11 line) and wherein the at least one of the plurality of

column oriented programming language statements is selectable (Figure 2, 203, Specification page 11 lines 18-21) in a first pane of a plurality of panes on a graphical user interface; (Figure 2, 200, 210, specification, page 10, lines 13-27) and, displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, (specification page 2 lines 27-29) and wherein the graphical user interface is adapted to receive content for the one or more editable fields from a user to define the at least one of the plurality of column oriented programming language statements. (Figure 2, 222, 224, 225, specification page 11 lines 4-21)

C. CLAIM 15 – INDEPENDENT

The subject matter of claim 15 is directed to a computer program product having a computer readable medium tangibly embodying computer executable code for directing a data processing system to edit a plurality of column oriented programming language statements presented to a user on a display screen. The computer program product comprises (Specification, page 3 lines 15-20) code for identifying a template description for at least one of the plurality of column oriented programming language statements, (Figure 4, Figure 5, reference numeral 410; specification page 1 lines 6-7, page 16, lines 17-26) wherein the template description defines at least one statement type, (Figure 2, reference numeral 221, specification page 2 line 18, page 11 lines 4-21) and wherein the at least one of the plurality of column oriented programming language statements is selectable (Figure 2, reference numeral 203, specification page 11 lines 18-21) in a first pane of a plurality of panes on a graphical user interface (Figure 2, reference numerals 200, 210, specification, page 10, lines 13-27); and code for displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements. (Figure 2, reference numerals 222, 224, 225, specification, page 11 lines 4-21)

D. CLAIM 22 – INDEPENDENT

The subject matter of claim 22 is directed to an article having a recordable type medium being usable over a network, and having means embedded in the recordable type medium for directing a data processing system to edit a plurality of column oriented programming language statements presented to a user on a display screen. The article comprises (Specification, page 3 lines 15-20) means in the recordable type medium (specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and (Figure 4, Figure 5, reference numeral 410, Figure 6 step 602 , specification page 22 lines 23-29, page 23 lines 24 through page 24 line 4) means in the recordable type medium (specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements. (Figure 6, steps 603, 604, 605, specification page 23 lines 1-22)

E. CLAIM 2 –DEPENDENT

The subject matter of claim 2 is directed to the method of claim 1, with the additional feature of permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement (Figure 2, reference numeral 232, page 12 lines 6-24).

F. CLAIM 3 –DEPENDENT

The subject matter of claim 3 is directed to the method of claim 2, with the additional feature of permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column

oriented programming language statements (Figure 2, reference numeral 232, page 12 lines 6-24).

G. CLAIM 4 –DEPENDENT

The subject matter of claim 4 is directed to the method of claim 3, with the additional feature of displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user (Figure 2, numeral 226, page 4 lines 4-12 and page 14 lines 1-4).

H. CLAIM 5 –DEPENDENT

The subject matter of claim 5 is directed to the method of claim 4, with the additional features of selectively replacing the at least one of the plurality of column oriented programming language statements and inserting the at least one of the plurality of column oriented programming language statements in the first pane (Figure 2, reference numeral 232, page 12 lines 6-24).

I. Claim 23 – DEPENDENT

The article of claim 22, further comprising:
means in the recordable type medium (Specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement (Figure 2, reference numeral 232, page 12 lines 6-24).

J. Claim 24 – DEPENDENT

The article of claim 23, further comprising:
means in the recordable type medium (Specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements (Figure 2, reference numeral 232, page 12 lines 6-24).

K. Claim 25 – DEPENDENT

The article of claim 24, further comprising:
means in the recordable type medium (Specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user (Figure 2, numeral 226, page 4 lines 4-12 and page 14 lines 1-4).

L. Claim 27 – DEPENDENT

The article of claim 25, further comprising means in the recordable type medium (Specification page 7 lines 4-18, page 23 line 24 through page 24 line 5) for one of: selectively replacing the at least one of the plurality of column oriented programming language statements; and
inserting the at least one of the plurality of column oriented programming language statements in the first pane (Figure 2, reference numeral 232, page 12 lines 6-24).

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION (Claims 1-28)

Whether claims 1-28 are obvious under 35 U.S.C. § 103 in view of *Reulein et al.* U.S. Patent Application Publication 2003/0142128, User Interface for a Document Component Management and Publishing System, (July 31, 2003) (hereafter “*Reulein*”) further in view of *Wygodny et al.* U.S. Patent Number 6,202,199, System and Method for Remotely Analyzing the Execution of Computer Programs (March 13, 2001) (hereafter “*Wygodny*”).

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1, 8, 15 and 22)

The ground of rejection is the assertion that claims 1, 8, 15 and 22 are obvious in view of *Reulein* and *Wygodny*. This rejection is in error and should be overturned.

A.1. Claims (1, 8, 15 and 22)

Claim 1 is a representative claim of this claim grouping. Claim 1 is as follows:

1. A method for editing a plurality of column oriented programming language statements presented to a user on a display screen, comprising:
 - identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and
 - displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements.

The Final Office Action of July 10, 2007 pp 3-4 states the rejection of claim 1 over 35 U.S.C. § 103 as follows:

Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Reulein et al.* (US200310142128A1) further in view of *Wygodny et al.* (US62021 99B1). Claims 1, 8, 15 and 22: *Reulein* discloses method, system and product for editing a plurality of column oriented programming language statements (*Reulein*: fig.7A and par. [0021]) presented to a user on a display screen (*Reulein*: fig.4: item# 202: "User Interface")), comprising: identifying a template description for at least one of the plurality of column oriented programming language statements (*Reulein*: fig.7A and par. [0021]), wherein the template description defines at least one statement type (*Reulein*: par. [0057], lines 3-11). While *Reulein* discloses a GUI (Fig. 3), *Reulein* does not disclose "a plurality of panes on a graphical user interface". However, *Wygodny* does disclose the at least one of the plurality of column oriented programming language statements is selectable in a first pane (*Wygodny*: fig.5 item #501; Column 12, lines

54-56) of a plurality of panes on a graphical user interface (Wygodny: fig.5; Column 12, lines 54-56); and displaying one or more editable fields corresponding to the at least one statement type on a second pane (Wygodny: fig.5 items #504; Column 12, lines 54-56) of the plurality of panes (Wygodny: i g.5 items #501 and 504; Column 12, lines 54-56) on the graphical user interface (Wygodny: fig.5), wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually (Wygodny: fig.5 item #501), and wherein the graphical user interface is adapted to receive content for the one or more editable fields from [[said.]] the user to define the at least one of the plurality of column oriented programming language statements (Wygodny: fig.5 item #504). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have a plurality of panes on a graphical user interface in Reulein's system. One would have been motivated to include a plurality of panes on a graphical user interface to show different option such as filter tree pane #501 and a source code pane #504 (Wygodny: fig.5: items #501 and #502; Column 12, lines 54-56).

The Office Action bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. In *re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. In *re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

A.1.i. The Proposed Combination Does Not Teach or Suggest All of the Features of Claim

1

The Office Action has failed to state a *prima facie* obviousness rejection against claim 1 because the proposed combination of *Reulein* and *Wygodny*, when considered as a whole, does not teach all of the features of claim 1. Specifically, the proposed combination does not teach the features of, “identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type.” The Office Action asserts *Reulein* teaches, “identifying a template description for at least one of the plurality of column oriented programming language statements (*Reulein*: fig.7A and par. [0021]), wherein the template description defines at least one statement type (*Reulein*: par. [0057], lines 3-11). While *Reulein* discloses a GUI (Fig. 3), *Reulein* does not disclose “a plurality of panes on a graphical user interface”.

Generally, *Reulein* is directed to a user interface for a document component management and publishing system and teaches a table containing a set of rules in Figure 7A. However, *Reulein* fails to teach, “identifying a template description for at least one of the plurality of column oriented programming language statements” because all of the rules of *Reulein* are defined in the table. For this reason, *Reulein* has no need to identify a template description to use with the statements as currently claimed. Further, as stated in the Office Action, *Reulein* fails to disclose, “a plurality of panes on a graphical user interface.” Therefore, *Reulein* fails to teach or suggest the features as currently claimed in claim 1.

Additionally, *Wygodny* also does not teach or suggest this claimed feature. Accordingly, neither reference teaches or suggests this claimed feature. Hence, the proposed combination of references, considered as a whole, fails to teach or suggest this claimed feature. Therefore, under the standards of *In re Royka*, the Examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

Additionally, *Wygodny* does not teach what the Examiner asserts *Wygodny* to teach. The Office Action asserts *Wygodny* teaches, “displaying one or more editable fields corresponding to the at least one statement type on a second pane (*Wygodny*: fig.5 items #504; Column 12, lines 54-56) of the plurality of panes (*Wygodny*: fig.5 items #501 and 504; Column 12, lines 54-56) on the graphical user interface (*Wygodny*: fig.5).”

Wygodny, however, displays only a “selection box” per statement, regardless of the type

of statement found in the first pane. Therefore, *Wygodny* neither teaches nor suggests displaying the “editable fields corresponding to the at least one statement type” as currently claimed.

The Office Action also asserts that *Wygodny* teaches, “the graphical user interface is used for editing the plurality of column oriented programming language statements individually (*Wygodny*: fig.5 item #501), and wherein the graphical user interface is adapted to receive content for the one or more editable fields from [[said.]] the user to define the at least one of the plurality of column oriented programming language statements (*Wygodny*: fig.5 item #504).” However, as explicitly stated in the abstract of *Wygodny*, *Wygodny* does not allow the editing of the programming language statements.

The Office Action states that *Reulein* does not teach or suggest these claimed features. Therefore the proposed combination of references, considered as a whole, does not teach or suggest these claimed features. Accordingly, the Examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

A.1.ii. Wygodny Teaches Away from the Invention of Claim 1

In addition, the Office Action has failed to establish a *prima facie* obviousness rejection against claim 1 because no reason exists under the standards of *KSR Int’l.* to combine the references, considered as a whole, in the manner suggested by the Office Action. No reason exists to combine the references because *Wygodny* teaches away from claim 1. A reference may be said to “teach away” from the claimed invention when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant. *In re Gurley*, 27 F.3d 551, 553, 31 U.S.P.Q.2D 1130, 1131 (Fed. Cir. 1995).

In this case, *Wygodny* is directed to a system and method for remotely analyzing the execution of computer programs. *Wygodny* teaches in the abstract that, “...tracing is performed without requiring modifications to the executable or source code files of the client.” For this reason, *Wygodny* prevents a user from modifying a source code statement. This teaching is diametrically opposed to the claimed feature of,

displaying one or more editable fields corresponding to the at least one

statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements

as recited in claim 1.

Thus, not only does *Wygodny* fail to teach, “editing the plurality of column oriented programming language statements individually,” as described above, but *Wygodny* explicitly prevents editing source code. Therefore, again, *Wygodny* teaches away from the features of “editable fields,” as claimed.

Additionally, while *Wygodny* teaches use of a selection box, the selection of a statement for tracing in *Wygodny* is not to be equated with the claimed feature of, “receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements,” because the selection of the statement fails to change the statement so as to create or define the statement, as is currently claimed. Thus, *Wygodny* still teaches away from claim 1.

Therefore, under the standards of *KSR Int’l.* and *In re Gurley*, no proper reason to achieve the legal conclusion of obviousness can be stated. Accordingly, the Examiner failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

A.1.iii. Impermissible Hindsight

The Office Action failed to state a *prima facie* obviousness rejection because the Office Action used impermissible hindsight when fashioning the rejections. "It is impermissible within the framework of section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art." *In re Hedges*, 228 U.S.P.Q. 685, 687 (Fed. Cir. 1986). Additionally, Personal opinion cannot be substituted for what the prior art teaches because a *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). In viewing

the references as a whole, one of ordinary skill would look to the problems addressed by the references in determining whether to combine the references.

As shown above, *Reulein* is directed to a user interface for a document component management and publishing system. In stark contrast, as shown above, *Wygodny* solves the wholly unrelated problem of remotely analyzing the execution of computer programs. The two problems are completely unrelated to each other.

Because no reason exists for one of ordinary skill to look to *Wygodny* to cure any lack of disclosure in *Reulein*, the Office Action incorrectly chose individual elements from *Wygodny* to support the rejection. This action is impermissible hindsight under the standards of *In re Hedges* and *In re Bell*. Therefore, the Office Action has failed to state a *prima facie* obviousness rejection against claim 1.

Based on the plain disclosures in the references, the only suggestion to modify the references is found in Applicants' specification. Hence, the Office Action must have used Applicants' specification to find a teaching, suggestion, or motivation to combine the references. Combining the references in this manner constitutes impermissible hindsight and fails to comport with the standards of *Graham v. John Deere Co.*, 383 U.S. 1 (1966), which requires a proper teaching, suggestion, or motivation to combine or modify references to achieve a proper obviousness rejection. Accordingly, the Office Action has failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

A.1.iv. No Proper Reason Stated Under KSR

Additionally, the Office Action failed to state a *prima facie* obviousness rejection against claim 1 because the Office Action failed to state a proper reason to combine the references under the standards of *KSR Int'l*. Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).

Regarding a reason to combine the references, the Office Action states that, "one would be motivated to include a plurality of planes on a graphical user interface to show different option such as filter tree pane #501 and a source code pane #504 (*Wygodny*: fig. 5 items #501, and #502 Column 12 lines 54-56)." However, this reason is not a rational underpinning to support the

legal conclusion of obviousness of claim 1 in view of the combination of the references considered as a whole. As a first matter, as shown above, *Reulein* does not teach use of the user interface with a plurality of panes and, “identifying a template description for at least one of the plurality of column oriented programming language statements.” As a second matter, *Wygodny* is directed towards a system and method for remotely analyzing the execution of computer programs, not displaying document management processes and restricting changing of source code. Therefore, the Office Action does not provide a rational underpinning to support the legal conclusion of obviousness, as required by *KSR Int’l*. Accordingly, under standards of *KSR Int’l*, the Office Action failed to state a *prima facie* obviousness rejection against claim 1 or any other claim in this grouping of claims.

Additionally, the Office Action provides no proper basis regarding a reason to reach the *legal conclusion* of obviousness. Again regarding a reason to combine the references, the Examiner states that:

one would be motivated to include a plurality of planes on a graphical user interface to show different option such as filter tree pane #501 and a source code pane #504 (Wygodny: fig. 5 items #501, and #502 Column 12 lines 54-56). One would have been motivated to include a plurality of panes on a graphical user interface to show different option such as filter tree pane #501 and a source code pane #504 (Wygodny: fig.5: items #501 and #502; Column 12, lines 54- 56).

Final office action of July 10, 2007, p. 4.

However, the Examiner’s statement does not satisfy the requirements of *KSR Int’l*. The Court in *KSR Int’l*. states that, “Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, *all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.*” *KSR Int’l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). “*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.*” *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).”

Additionally, the Examiner's statement provides *no rational underpinning to support the legal conclusion of obviousness*. The Examiner only states a purported advantage to combine the references (to show different options such as filter tree pane and source code pane). However, the Examiner does not provide any technical or other rational connection between the purported advantage *and the legal conclusion* of obviousness. Instead the Examiner simply states the purported advantage and then assumes that the reader would recognize that the purported advantage would somehow compel the *legal conclusion* that claim 1 is obvious in view of the references. However, this assumption fails to comport with the requirements of *KSR Int'l* that the Examiner provide articulated reasoning with a rational underpinning to achieve the legal conclusion of obviousness in view of the references.

Therefore, the Examiner's statement is both conclusory and provides no articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. Accordingly, under the standards of *KSR Int'l.*, the Examiner failed to state a prima facie obviousness rejection against claim 1 or any other claim in this grouping of claims.

A.1.v. The Proposed Combination Changes the Principle of Operation of the Primary Reference

Further regarding claim 1, the Office Action has failed to state a prima facie obviousness rejection because the proposed combination changes the principle of operation of the primary reference. In combining references to show the claimed feature, the proposed modification cannot change the principle of operation of a reference. See *In re Ratti*, 270 F.2d 810, 123 (CCPA 1959) and MPEP 2143.01. If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious. *Id.*

In the case at hand, the proposed combination of the references changes the principle of operation of *Wygodny*. *Wygodny* is directed towards a system and method for remotely analyzing the execution of computer programs, not displaying document management processes and preventing the changing of source code.

The Office Action's proposed combination changes the principle of operation of *Wygodny* because, in the Office Action's proposed combination of remotely analyzing the

execution of computer programs in which no source code changes are allowed with the document, the system of *Reulein* would force *Wygodny* to alter source code. Thus, the display of *Wygodny* cannot be used as suggested by the Office Action because it causes *Wygodny* to be inoperative. Accordingly, the Office Action's proposed combination changes the principle of how the data is managed in *Wygodny*, contrary to the very teaching of *Wygodny*.

As shown above, *In re Ratti* provides that changing the principle of operation of a reference renders a claim non-obvious in view of the proposed combination. Therefore, claim 1 is non-obvious in view of the proposed combination, as are all other claims in this grouping of claims.

A.2. GROUND OF REJECTION 2 (Claims 2, 9, 16 and 23)

Claim 2 is a representative claim of this claim grouping. Claim 2 is as follows:

The method of claim 1, further comprising:
 permitting the at least one of the plurality of column oriented
 programming language statements to be selected by the user for
 replacement.

The Final Office Action of July 10, 2007 page 4 states the rejection of claims 2, 9, 16 and 23 over 35 U.S.C. § 103 as follows:

Claims 2, 9, 16 and 23: *Reulein* in view of *Wygodny* discloses method, system and product of claims 1, 8, 15, & 22 respectively and further comprising permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement (*Reulein*: Paragraph [0042], lines 6-12).

A.2.i. The Proposed Combination Does Not Teach or Suggest All the Features of Claim 2

The Office Action has failed to state a *prima facie* obviousness rejection against claim 2 because the proposed combination of *Reulein* and *Wygodny*, when considered as a whole, does not teach all of the features of claim 2. Specifically, the proposed combination does not teach the features of, "permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement." *Reulin*, as stated previously, is directed to a user interface for a document component management and publishing system. The cited portion of *Reulein* teaches an author in box from which tasks can be selected as in:

From within the in-box, each author user can select individual tasks, manage assigned documents and document components which allows the user to check in or out, edit and approve or reject documents or document components, all within the workflow. Any modified, new, approved or rejected documents or document components are routed to an appropriate user (author) for subsequent action, e.g., revision, correction, review, etc

The cited reference teaches task selection only and fails to teach replacing tasks in the in box. The user must go to other part of the system of *Reulein* to work on the documents. Previously the Office Action equated the table of rules of *Reulein* to the column oriented programming language statements to be selected by the user as currently claimed.

The Office Action appears to be equating the tasks of the author in box to the same column oriented programming language statements to be selected by the user. However, even if this assertion were correct, *Reulein* fails to teach replacement of a task as is currently claimed.

As stated previously, *Wygodny* is directed to a system and method for remotely analyzing the execution of computer programs. *Wygodny* teaches in the abstract that, "...tracing is performed without requiring modifications to the executable or source code files of the client." For this reason, *Wygodny* prevents a user from modifying a source code statement. This teaching is diametrically opposed to the claimed feature of, "permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement."

Therefore the proposed combination of references of the Office Action, considered as a whole, does not teach or suggest the claimed feature. Additionally, for the reasons stated above, the Office Action failed to state a *prima facie* obviousness rejection against claim 1, from which claim 2 depends. Accordingly, the Office Action failed to state a *prima facie* obviousness rejection against claims 2, or any other claim in this grouping of claims.

A.3. GROUND OF REJECTION 3 (Claims 3, 10, 17 and 24)

Claim 3 is a representative claim of this claim grouping. Claim 3 is as follows:

The method of claim 2, further comprising:
 permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.

The Final Office Action of July 10, 2007 page 4 states the rejection of claims 3, 10, 17 and 24 over 35 U.S.C. § 103 as follows:

Claims 3, 10, 17 and 24: *Reulein* in view of *Wygodny* discloses method, system and product of claims 2, 9, 16, & 23 respectively and further comprising permitting a position in the plurality of column oriented programming language statements to be selected by [[said]] the user for insertion of the at least one of the plurality of column oriented programming language statements (*Reulein*: Paragraph [0042], lines 6-14).

A.3.i. The Proposed Combination Does Not Teach or Suggest All the Features of Claim 3

The Office Action has failed to state a *prima facie* obviousness rejection against claim 3 because the proposed combination of *Reulein* and *Wygodny*, when considered as a whole, does not teach all of the features of claim 3. Specifically, the proposed combination does not teach or suggest the features of, “permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.” *Reulein*, as stated previously, is directed to a user interface for a document component management and publishing system. The cited portion of

Reulein teaches:

From within the in-box, each author user can select individual tasks, manage assigned documents and document components which allows the user to check in or out, edit and approve or reject documents or document components, all within the workflow. Any modified, new, approved or rejected documents or document components are routed to an appropriate user (author) for subsequent action, e.g., revision, correction, review, etc. Also, users can add notes to a particular object within a project through the in-box 1540.

The cited reference teaches task selection only, but fails to teach insertion of tasks in the “in box.” The user must go to other part of the system of *Reulein* to work on the documents. Previously, the Office Action equated the table of rules of *Reulein* to the column oriented programming language statements to be selected by the user as currently claimed. Now, in an apparent self-contradiction, the Office Action appears to be equating the tasks of the author “in box” to the same column oriented programming language statements to be selected by the user. Even if this assertion were correct, *Reulein* fails to teach insertion of a task, as claimed.

As stated previously, *Wygodny* is directed to a system and method for remotely analyzing the execution of computer programs. *Wygodny* teaches in the abstract that, "...tracing is performed without requiring modifications to the executable or source code files of the client." For this reason, *Wygodny* prevents a user from modifying a source code statement. This teaching is diametrically opposed to the claimed feature of, "permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements."

Therefore, the proposed combination of references of the Office Action, considered as a whole, does not teach or suggest the claimed feature. Additionally, for the reasons stated above, the Office Action failed to state a *prima facie* obviousness rejection against claim 1, from which claim 3 depends. Accordingly, the Office Action failed to state a *prima facie* obviousness rejection against claims 3 or any other claim in this grouping of claims.

A.4. GROUND OF REJECTION 4 (Claims 4, 11, 18 and 25)

Claim 4 is a representative claim of this claim grouping. Claim 4 is as follows:

The method of claim 3, further comprising:
displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.

The Final Office Action of July 10, 2007 page 5 states the rejection of claims 4, 11, 18 and 25 over 35 U.S.C. § 103 as follows:

Claims 4, 11, 18 and 25: *Reulein* in view of *Wygodny* discloses method, system and product of claims 3, 10, 17, & 24 respectively and further comprising displaying a field difference indicator on the graphical user interface for each of [[said]] the one or more editable fields whose contents has been changed by [[said]] the user (*Reulein*: Paragraph [0032], lines 5-8).

A.4.i. The Proposed Combination Does Not Teach or Suggest All the Features of Claim 4

The Office Action has failed to state a *prima facie* obviousness rejection against claim 4 because the proposed combination of *Reulein* and *Wygodny*, when considered as a whole, does not teach all of the features of claim 4. Specifically, the proposed combination does not teach or suggest the feature of, "displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user." *Reulin*, as

stated previously, is directed to a user interface for a document component management and publishing system. The cited portion of *Reuelin* paragraph [0032] lines 5-8, teaches, “the preferred embodiment system provides a graphical representation with different colors and boxes depicting the current status of the particular project. Each project workflow is configurable on a client-by-client basis.” Use of color indicating the status of a project does not provide the type of indicator as is currently claimed. The teaching of *Reulein* fails to provide an indication at the field level of a change as is currently claimed. *Reulein* fails to indicate what in the project may have changed so as to alert the user in the same manner as is claimed. Therefore *Reulein* fails to teach the feature as claimed.

As stated previously, *Wygodny* is directed to a system and method for remotely analyzing the execution of computer programs. *Wygodny* teaches in the abstract that, “...tracing is performed without requiring modifications to the executable or source code files of the client.” For this reason, *Wygodny* prevents a user from modifying a source code statement. This teaching is diametrically opposed to the claimed feature of, “displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.” In the teaching of *Wygodny*, the user is prevented from making a source code change, and therefore *Wygodny* has no need of providing such an indicator, as claimed.

Therefore, the proposed combination of references of the Office Action, considered as a whole, does not teach or suggest the claimed feature. Additionally, for the reasons stated above, the Office Action failed to state a *prima facie* obviousness rejection against claim 1, from which claim 4 depends. Accordingly, the Office Action failed to state a *prima facie* obviousness rejection against claims 4, or any other claim in this grouping of claims.

A.5. GROUND OF REJECTION 5 (Claims 5, 12, 19 and 26)

Claim 5 is a representative claim of this claim grouping. Claim 5 is as follows:

The method of claim 4, further comprising one of:
selectively replacing the at least one of the plurality of column oriented programming language statements; and
inserting the at least one of the plurality of column oriented programming language statements in the first pane.

The Final Office Action of July 10, 2007 page 5 states the rejection of claims 5, 12, 19 and 26 over 35 U.S.C. § 103 as follows:

Claims 5, 12, 19 and 26: *Reulein* in view of *Wygodny* discloses method, system and product of claims 4, 11, 18 & 25 respectively and further comprising one of: selectively replacing the at least one of the plurality of column oriented programming language statements; and inserting the at least one of the plurality of column oriented programming language statements in the first pane (*Wygodny*: Column 21, lines 11-17).

A.5.i. The Proposed Combination Does Not Teach or Suggest All the Features of Claim 5

The Office Action has failed to state a *prima facie* obviousness rejection against claim 4 because the proposed combination of *Reulein* and *Wygodny*, when considered as a whole, does not teach or suggest all of the features of claim 5. Specifically, the proposed combination does not teach or suggest the features of, “selectively replacing the at least one of the plurality of column oriented programming language statements; and inserting the at least one of the plurality of column oriented programming language statements in the first pane.”

As stated previously, *Wygodny* is directed to a system and method for remotely analyzing the execution of computer programs. *Wygodny* teaches in the abstract that, “...tracing is performed without requiring modifications to the executable or source code files of the client.” For this reason, *Wygodny* prevents a user from modifying a source code statement. This teaching is diametrically opposed to the claimed feature of, “selectively replacing the at least one of the plurality of column oriented programming language statements; and inserting the at least one of the plurality of column oriented programming language statements in the first pane.” In the teaching of *Wygodny*, the user is prevented from making a source code change and therefore *Wygodny* has no need of providing such a selective replacement and insertion functions, as claimed.

In the cited passage, *Wygodny* teaches selection of an object in a source code view causing the statements being viewed to change to display another portion of the same source code. Specifically, *Wygodny* teaches at col. 21 lines 11-17:

In this case, the developer 112 can change the source file search path in order to display the source code. To change the source file path the developer should select a function in the trace tree 330, then right-click in the source pane to open a pop-up menu, and then select the "Source Location" from the pop-up menu. Alternatively, the developer 112 can add additional source directories and remove source directories by selecting

the "Options" command from the "View" menu on the menu bar 304. Source file paths can also be removed.

In the cited passage, *Wygodny* teaches movement within a source code file by selecting an object rather than selectively replacing and inserting statements as currently claimed. Further in the teaching of *Wygodny*, a selection in the first pane containing trace tree 330 causes a change in the source code viewed in the source pane. There is no change in the trace tree 330 due to a change in the source pane as taught by *Wygodny*. Regardless of which direction a change may be directed in *Wygodny*, *Wygodny* does not teach selectively replacing and inserting statements, as claimed. *Wygodny* therefore fails to teach the claimed features of claim 5.

The Office Action states that, "While *Reulein* discloses a GUI (Fig. 3), *Reulein* does not disclose "a plurality of panes on a graphical user interface". Therefore *Reulein* cannot be used to provide the missing features from *Wygodny* to achieve the claimed features of claim 5 as currently claimed. The combination of *Reulein* and *Wygodny* therefore fails to provide the claimed features of claim 5. For this reason, the proposed combination of references of the Office Action, considered as a whole, does not teach or suggest these claimed features. Additionally, for the reasons stated above, the Office Action failed to state a *prima facie* obviousness rejection against claim 1, from which claim 5 depends. Accordingly, the Office Action failed to state a *prima facie* obviousness rejection against claim 5 or any other claim in this grouping of claims.

B. CONCLUSION

As shown above, the Office Action has failed to state valid rejections against the claims. Therefore, Applicants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Applicants request the Board to direct the Examiner to allow the claims.

/Theodore D. Fay III/
Theodore D. Fay III
Reg. No. 48,504
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method for editing a plurality of column oriented programming language statements presented to a user on a display screen, comprising:

identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and

displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements.

2. The method of claim 1, further comprising:

permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement.

3. The method of claim 2, further comprising:

permitting a position in the plurality of column oriented programming language

statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.

4. The method of claim 3, further comprising:

displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.

5. The method of claim 4, further comprising one of:

selectively replacing the at least one of the plurality of column oriented programming language statements; and

inserting the at least one of the plurality of column oriented programming language statements in the first pane.

6. The method of claim 5, further comprising:

displaying a user selectable apply button on the graphical user interface for initiating one of the selectively replacing and the inserting.

7. The method of claim 1, wherein the template description is an extensible mark-up language document.

8. A system for editing a plurality of column oriented programming language statements, the system having a display and an input device for editing the plurality of column oriented programming language statements through a graphical user interface, the system comprising:

a processor coupled to the display and the input device and the processor adapted for identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and, displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, , and wherein the graphical user interface is adapted to receive content for the one or more editable fields from a user to define the at least one of the plurality of column oriented programming language statements.

9. The system of claim 8, wherein the processor is further adapted for permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement.

10. The system of claim 9, wherein the processor is further adapted for permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.

11. The system of claim 10, wherein the processor is further adapted for displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.
12. The system of claim 11, wherein the processor is further adapted for one of:
selectively replacing the at least one of the plurality of column oriented programming language statements; and
inserting the at least one of the plurality of column oriented programming language statements in the first pane.
13. The system of claim 12, wherein the processor is further adapted for displaying a user selectable apply button on the graphical user interface for initiating one of the selectively replacing and the inserting.
14. The system of claim 8, wherein the template description is an extensible mark-up language document.
15. A computer program product having a computer readable medium tangibly embodying computer executable code for directing a data processing system to edit a plurality of column oriented programming language statements presented to a user on a display screen, the computer program product comprising:
code for identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one

statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and

code for displaying one or more editable fields corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements.

16. The computer program product of claim 15, further comprising:

code for permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement.

17. The computer program product of claim 16, further comprising:

code for permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.

18. The computer program product of claim 17, further comprising:

code for displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.

19. The computer program product of claim 18, further comprising code for one of:
selectively replacing the at least one of the plurality of column oriented programming language statements; and
inserting the at least one of the plurality of column oriented programming language statements in the first pane.
20. The computer program product of claim 19, further comprising:
code for displaying a user selectable apply button on the graphical user interface for initiating one of the selectively replacing and the inserting.
21. The computer program product of claim 15, wherein the template description is an extensible mark-up language document.
22. An article having a recordable type medium being usable over a network, and having means embedded in the recordable type medium for directing a data processing system to edit a plurality of column oriented programming language statements presented to a user on a display screen, the article comprising:
means in the recordable type medium for identifying a template description for at least one of the plurality of column oriented programming language statements, wherein the template description defines at least one statement type, and wherein the at least one of the plurality of column oriented programming language statements is selectable in a first pane of a plurality of panes on a graphical user interface; and
means in the recordable type medium for displaying one or more editable fields

corresponding to the at least one statement type on a second pane of the plurality of panes on the graphical user interface, wherein the graphical user interface is used for editing the plurality of column oriented programming language statements individually, and wherein the graphical user interface is adapted to receive content for the one or more editable fields from the user to define the at least one of the plurality of column oriented programming language statements.

23. The article of claim 22, further comprising:

means in the recordable type medium for permitting the at least one of the plurality of column oriented programming language statements to be selected by the user for replacement.

24. The article of claim 23, further comprising:

means in the recordable type medium for permitting a position in the plurality of column oriented programming language statements to be selected by the user for insertion of the at least one of the plurality of column oriented programming language statements.

25. The article of claim 24, further comprising:

means in the recordable type medium for displaying a field difference indicator on the graphical user interface for each of the one or more editable fields whose contents has been changed by the user.

26. The article of claim 25, further comprising means in the recordable type medium for one of:

selectively replacing the at least one of the plurality of column oriented programming

language statements; and

inserting the at least one of the plurality of column oriented programming language statements in the first pane.

27. The article of claim 26, further comprising:

means in the recordable type medium for displaying a user selectable apply button on the graphical user interface for initiating one of the selectively replacing and the inserting.

28. The article of claim 22, wherein the template description is an extensible mark-up language document.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.